TONAL DisCo: DISSONANCE AND CONSONANCE IN A GAMING ENGINE

Vinh Xuan Nguyen

Faculty of Built Environment,
University of New South Wales
Sydney NSW 2052, Australia
vinh.x.nguyen@unsw.edu.au

ABSTRACT

Whilst there are several existing toolkits specifically designed for sonification, there has been little investigation into the utilization of computer game engines for sonification. This paper will demonstrate the implementation of a real time game engine for the purpose of sonification and discuss the opportunities and limitations.

An important aspect which is lacking in existing sonification toolkits is the ability to sonify streaming data in real-time. Gaming engines not only offer the potential to do this but also offer the ability to visualize data in 3D and in real-time.

The sound design of an art exhibition is used as a case study to demonstrate the potential of a computer game editor/sandbox used for visualization and sonification. For the exhibition real-world objects were tracked inside a gallery space and represented in the virtual environment of a computer game, which was displayed on a projector screen. Their movement was sonified into musical form to convey their steady/consistent movement as consonant and their agitated/inconsistent movement as dissonant. Tonal "DisCo" is used to describe their dissonance and consonance rating in both musical tone and visual color.

Although the sonification of data into musical structure distorts the accuracy of absolute data values, it does maintain the relationship between data values. This loss of resolution is counteracted by an increase in clarity of data relationships. This case study appropriates the single ratio scale of pitch into both an interval scale of tone and an ordinal scale of octaves in order to express interrelationships.

1. INTRODUCTION

There is a need for accessible sonification tools that offer flexible control over variables for parametric mapping [1]. Furthermore there is a need for sonification tools that offer real-time interactivity. Whilst there are several existing toolkits specifically designed for sonification, there has been little investigation into the utilization of computer game engines for sonification.

There are existing tools and toolkits specifically designed for sonification. Generally these offer customization of sound parameters (pitch, volume, timbre etc.) for sonification as well as visualization in 2D. LISTEN was written in C++ and designed with logical modules for portability [2]. Lodha points out that "the sounds generated [in LISTEN] are non-musical [and] they can be fatiguing when exploring large data sets over extended periods of time." [3] This prompted its evolution into MUSE similarly written in C++ but instead implementing musical sounds. The development of MUSART [4] made a significant contribution to already existing tools by producing "musical sound maps" that could be played in real-time. In addition one of the small but important contributions was the designation of silence or "rests," as they are called in music notation. Walker's Sonification Sandbox [5] is a newer toolkit that is written in Java and is intended as a cross-platform program. It allows data to be imported as CSV files (comma separated values), utilizes a user friendly interface with intuitive slider bars and is able to generate MIDI files which can be imported by different audio programs. Although it offers visualization capabilities, it is limited to graphing on a 2D plane. The key aspect which is missing from most of these existing toolkits is the ability to sonify streaming data in real-time.

In addition to sonification toolkits, there are several programming environments which are directed at sound synthesis and production. These include Pure Data (PD) developed by Miller Puckette[1], Max by Cycling 74[2], and Impromptu[3]. Some of these offer 3D visualization capabilities. PD and Max are two programming environments which enable users to construct logic using graphics rather than code. They both, however still require knowledge of code keywords and arguments. This need for coding knowledge limits the accessibility to a wider user base and inhibits a user from approaching the program logic from a design-orientated perspective. Extensions such as MSP and Jitter have been developed for Max which allow audio synthesis/processing and 2D/3D graphics using OpenGL respectively. Computer game engines already offer advanced and complex visualization in 3D without the need to write programming logic. Impromptu, a text programming environment, has the key advantage of being a continuously running environment that does not require being rebuilt/cooked/compiled. So changes to the code can be made and tested in real time. However the program is still limited to users who want to operate in a code environment. These programming environments offer an ability to sonify data in real-time and in the case of PD and Max, offer interactive visual scripting capabilities, however they still use code keywords and do not offer an opportunity to visualize in 3D to the same level as real time 3D game engines.

[1] http://puredata.info/
[2] http://cycling74.com/products/maxmspjitter/
[3] http://impromptu.moso.com.au/

Gaming engines not only offer the ability to sonify data in real-time but also offer the innate ability to visualize in 3D and in real-time. The toolkits which are commonly distributed with computer games, usually in the form of an editor/sandbox, allow end-users to modify existing content and create custom content. These editors, which are commonly used for computer game modification and serious simulation and visualization, have the potential to be used for sonification.

This paper will demonstrate the implementation of a gaming engine, through the use of its editor/sandbox, for sonification and discuss the opportunities and limitations.

2. BACKGROUND

Relevant to this paper are the intersections between gaming and sonification, music and sonification, and dissonance and consonance.

2.1. Gaming and Sonification

There are many contemporary computer games which feature changes in sound effects and musical modes that are controlled by a player's actions. There are some that have been designed specifically to allow players to manipulate sound environments.

"de Blob" (on platforms: Nintendo Wii, Play Station 3, XBox 360, Nintendo DS and IPhone) is a game in which graphic and music is manipulated by the user. The player/composer "not only has control over the visual environment, but also the soundscape." [6] This partnership between visual and audio as well as interactivity in 3D is an advantage gaming engines offer over existing sonification toolkits. However "de Blob" does not allow customization of game content, sound content or sound variables.

On the other hand, there are some cases where computer games openly offer a way to create custom content. Second Life has been modified, by end-users, to be utilized for a purpose for which they were not intended, including sonification. The art-projects by Robinson [1] in Second Life demonstrate the potential of gaming engines when they are repurposed to perform tasks which they were not designed to do by developers. The interactivity becomes an essential element for "play" in the sense that the sonification systems designed only operate through interactive behavior with user controlled avatars. Computer games inherently lend themselves to one of the important aspects of sonification, interactivity [7,8].

First Person Shooter (FPS) game engines have already been compared to sonification systems [9]. When a FPS game is used in a way its developers intended, it is already acting as a sonification system. It is sonifying the actions of the player (even if they are as mundane as finger presses) and their interaction with a virtual environment through the embodiment of an avatar. This paper expands on the sonification of embodied avatars, but instead embodiment through tracking technology rather than keyboard or controller.

LeGroux's VR-RoBoser, created in a gaming engine called Torque3D, is a development of a virtual environment whose sonic representation is manipulated by avatars and their behavior [10]. Like Grimshaw, LeGroux perceives computer gaming as a real-time sonification system but uses data created by the system itself (the avatar). LeGroux alludes to the possibility of using an external data set, stating the potential to "sonify the interaction between physically present humans, virtual humans and synthetic characters." The realization of this possibility is demonstrated in this paper where the sonification of an external data set that is created outside the environment of the game, and as such is not restrained or limited by its rules and interface.

Other computer game engines which have had little attention as sonification systems are Crytek's CryEngine2 and Epic's Unreal Engine 3 engine. Both offer visual scripting capabilities, in the form of "Flow Graph" and "Kismet" respectively, allowing for flexible and design-orientated programming. This enables designers to manipulate data flow in real-time and influence elements within the game, such as visualization and sonification.

2.2. Music and Sonification

The difference between musical tone and noise is uniformity and structure. Helmholtz stated that "noise is accompanied by a rapid alternation of different kinds of sensations of sound." Whilst "on the other hand, a musical tone strikes the ear as a perfectly undisturbed, uniform sound" [11]. This perception of differentiating between musical tone and noise does not explicitly determine whether effective sonification should be constructed from tones or noises, but does help to direct a classification of a sonification as a whole.

If a sonification has the need to communicate high resolution data with a multitude of increments and changes, then it should take the form of full spectrum noise. If, on the other hand, a sonification is intended to communicate patterns and trends, then it should take the form of a musical structure so that a listener can discern and analyze the structure for relationships and irregularities. This is also subject-dependent, since a listener may assess some music structures as "noisy", and likewise some noises to as "musical".

The theory of John Cage's "indeterminacy" gives merit to a real-time interaction between composer and composition [12]. The unfolding of music over time becomes a process where each performance is unique even though it is composed from the same "score" or set of rules. Sonification mappings and their interactivity in real-time allow a similar composition in an indeterminate environment. The implementation of a game engine editor may provide an opportunity to promote the user to the status of composer rather than performer.

A more structured approach to music can be seen in Xenakis who systematically developed simple grains of complex sound into musical screens [13] and then developed those into works of music. For Xenakis music can be built up in texture from music fragments created by rigorous rules. Musical structure helps granulate sound into perceivable increments for a listener, and imposing music structure onto fine data so that it becomes coarser in grain, enables relational information to be perceived by a listener.

The argument for using a formal music structure for sonification has been established [14] and has been

[1] http://www.annamorphic.co.uk/

demonstrated in program debugging with the CAITLIN system [15]. Being able to discern programming "bugs" from musical cues has the advantage of alarming a listener by straying away from what becomes an expected structure.

The use of musical structure in sonification has the added potential to communicate compound relationships. These are relationships which are simultaneously similar and different. The dimension of pitch can be divided into a set of octaves and a subset of tones. Although the full spectrum is now compressed to only 12 tonal values, the expressed similarities between other octaves give a more accurate picture of its relationship to the other values.

There has been a call for consideration when using the term sonification for the parametric mapping of data to musical form with an underlying interest in aesthetics to produce pleasant sound. The term soniculation [16] gives specificity to the intention to maintain clarity of information. This type of sonification outlined in this paper is aligned with the term soniculation even though it attempts to sonify data into a systematic musical structure, because it does so to improve the communication of relational information. Aesthetics in this case becomes a mechanism to provide clarity, because it is the discord between data relationships (that is its dissonance) which enables a listener to recognize differences, changes and trends.

2.3. Dissonance and Consonance

Dissonance and consonance, as components of music, have been examined and theorized by others. Some of the more influential work is introduced here along with their implications for mapping a spectrum of tonal DisCo in sonification. The details of the how streaming data is translated into tonal DisCo is covered in section 4.3.

Early work (1863) by Helmholtz explains how pairs of tones when heard simultaneously were perceived as dissonant due to a disturbance caused by "beats" between their harmonics [11]. This concept clearly shows the resonating aspect of dissonance, whereby a single note due to its harmonics can be deemed related or opposed to every other note on a musical scale.

Later work (1898) by Stumpf used a concept of "tonal fusion" to describe how two tones can be heard as one. Those tones that lent themselves to fusion were considered more consonant that others [17,18]. To adopt this concept of dissonance is to also acknowledge a wider spectral band of consonance to dissonance that is much more difficult to grasp for a listener. If we take the key of C, then two notes of C several octaves apart may "fuse" better than a perfect fifth only less than one octave away.

Relatively recent work (1965) by Plomp and Levelt critically analyzes the work of their predecessors, Helmholtz and Stumpf, to develop a theory of "critical bandwidth." [19] They explain how dissonance is affected by a frequency difference between tones. The elegance of this concept enables a dissonance spectrum within a single octave than can be transposed to all octaves. The implication is a succinct tonal spectrum (from 1-12) that may be paired up with a specified octave, to become an identifier for a single note.

The work of Schoenberg in some ways opposes all of the concepts above. By employing a 12-tone-technique in which there was equality of all tones Schoenberg developed a style that "treats dissonances like consonances and renounces a tonal centre" [20]. In 1926 he used the phrase "emancipation of dissonance" which actually acknowledges the importance of dissonance in music. Schoenberg later "defined consonances as 'the more obvious, simpler relations to a fundamental', [and] dissonances as 'the more remote, more complicated ones.'" [20] These complicated relations are arguably of value to sonifications that convey multi-variant information.

A classic arrangement of dissonance and consonance exists in the circle of fifths, which as Jensen observes, first appears in Nikolai Diletskii's "Grammatika" [21]. The advantage that this presents over other theories is the closed-loop spectrum which can be employed to suggest a direction of movement between dissonant and consonant poles. This is a feature that is missing from other linear spectrums.

Tonal "DisCo" is used to describe the dissonance and consonance rating of a tone. It is a 12 step spectrum aligned with the 12 notes of the western musical scale. Tonal DisCo seeks to expand the use of the pitch spectrum in sonification by limiting the comparison of a tone, to those notes within its octave. Doing so allows a tone to carry information represented by its dissonance rating and its octave.

3. EXHIBITION SOUND DESIGN AS CASE STUDY

This paper discusses the implementation of a game engine for sonification and uses the sound design of an art exhibition as a case study for mapping dissonance and consonance in the movement of bodies.

A modified computer game engine was utilized for an exhibition in the "Beijing Today Art Museum"[1]. Real-world objects were tracked inside the gallery space and represented in the virtual environment of the game, which was displayed on a projector screen. The virtual environment became a testing ground for different parametric mappings of dissonance and consonance.

The sound design proposed using a dissonance rating based on data extrapolated from the movement of bodies, in this case the real-world objects. It is important to distinguish between body movement and the movement of a body. The former suggests the movement of individual components of a body (its arms, hands, legs, torso, head etc.) whilst the latter generalizes the single movement of a whole body. This research identifies the latter movement of a body as a source of data that can be sonified in real-time through the platform of a modified computer game.

In the sound design of this case study, a body refers to a tracked object in the gallery. This particular exhibition in Beijing tracks a wheelchair and a scooter and virtualizes their real world movement.

---

[1] *Magic Spaces*, curator Binghui Huangfu, 2nd floor, building 1, Today Art Museum, 32 Baiziwan Road, Chaoyang District, Beijing, China, 25 Feb - 20 Mar, 2011. http://www.todayartmuseum.com/en/Exhibition/eExhibitDetail.aspx?ActionType=0&Exhit=3&ChannelID=460&ExhibitID=3483 [accessed 28/2/2011]

4. IMPLEMENTATION AND PROCEDURE

The implementation of a gaming engine for sonification is outlined here with specific details describing preparation, the gaming engine editor, the mapping of tonal DisCo, and installation.

4.1. Preparation

The use of a game engine for sonification requires the use of a sound library compatible with the game engine. A sound library was created from sound samples in ".wav" format and built using FMod, a sound compression program designed for game engines. In a gaming engine, sounds are ironically not generated in real-time since this requires processing time. They are instead called on to be played as sound events from pre-recorded sound samples to reduce latency and ensure speed in a real-time system. This enables interaction between several sound events and most importantly allows streaming data to be handled in real-time.

4.2. Gaming Engine

The gaming engine used was Crytek's CryEngine 2 and the editor used was Crysis Wars Sandbox 2[1]. The driving factor behind this selection was accessibility and graphic capabilities. It would have been possible to utilize a different engine such as Epic's Unreal Engine 3 through the use of the Unreal Editor[2], however older engines such as UT2004[3] do not offer a visual scripting editor.

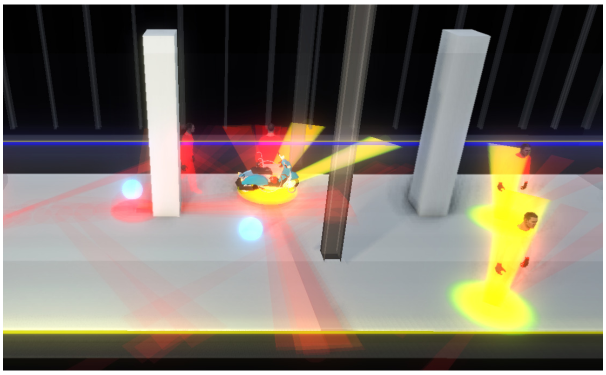

Figure 1: Inside the game: colors showing dissonant rating derived from the movement of bodies.

"Flow Graph" is a visual scripting tool within the Crysis Wars Sandbox 2 which "can be used to prototype gameplay, effects or sound design." and allows a designer to build "simple and complex logic." [22]

The main use of the Flow Graph in this design case was to call sound events from a compressed library, harvest numerical data from a data stream, position and rotate objects, and alter textures[4]. In addition it was used to periodically retrieve data from an external source and perform real-time calculations on any incoming data. This included scaling and translating values into ranges appropriate for use as sound parameters.

4.3. Color and Dissonance

The dissonance and consonance in this case study are extended to visualization in the form of color designation. In a similar method to Malinowski [23], the circle of fifths, originating from Diletskii's Grammatika [21] is imposed on a color wheel which intersects two color systems (RGB and CMYK) to make a cyclical spectrum of 12 colors. (Fig. 2) This ensures that the movement of a body is represented in a visual and sonic form that is consistent with its dissonance rating.

For example, in one experimental mapping, the time spent at a constant velocity returned a value above or equal to zero seconds. This value was altered in Flow Graph to an integer and given a maximum limit of seven seconds (0-7). This spectrum was remapped by taking the absolute value of the limited integer less 12. The result (12, 11, 10…5) referred to a color code of light blue to red (see Fig. 1).

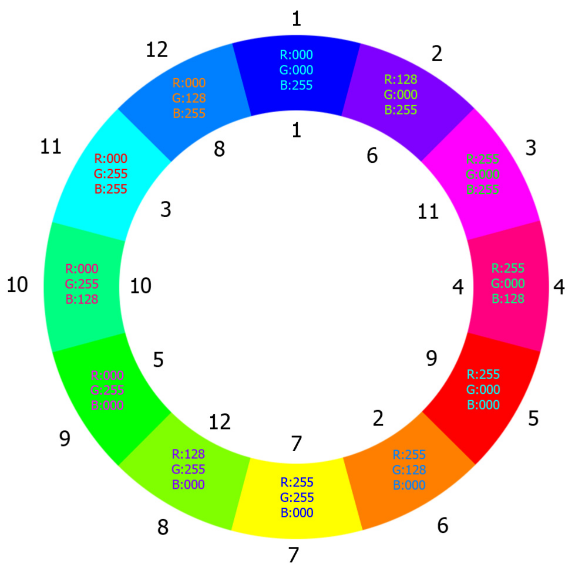

Figure 2: Chromatic circle (outer values) circle of fifths (inner values) and RGB values.

4.4. Dissonance and Consonance

This case study appropriates the single *ratio* scale of pitch into both an *interval* scale of tone and an *ordinal* scale of octaves in order to express interrelationships. These terms have been borrowed from Stevens [24]. Instead of a full linear spectrum of

[1] http://crytek.com/cryengine/cryengine2/overview
[2] http://www.unreal.com/technology.php
[3] http://www.beyondunreal.com/main/ut2004/ut2004essential.php

[4] http://fgps.sourceforge.net. Utilising a 3rd party plugin called "Flow Graph Plugin System" the "Flow Graph" can be used to retrieve and store data in XML files (a simple UTF-8 text file organized with <tags> to create a navigation friendly structure).

pitch in Hz, data is mapped to a granular DisCo rating of 1-12, while octaves are used to convey data category.

The importance of relational information has been established by Zhang [25] and although the sonification of data into musical structure distorts the accuracy of absolute data values, it does maintain the relationship between data values. This loss of resolution is counteracted by an increase in clarity of data relationships.

4.5. Mappings

From the circle of fifths discussed earlier [21] there is a cyclical order of consonance and dissonance that can be transposed to any key. Table 1 shows both clockwise and counter clockwise mappings. Each number below refers to a note on the western music scale, where 1 is the keynote/unison and therefore the most consonance, whilst 7 on the opposing side of the circle is the most dissonant.

Linear scale	1	2	3	4	5	6	7	8	9	10	11	12
Cyclic ccw	1	8	3	10	5	12	7	2	9	4	11	6
Cyclic cw	1	6	11	4	9	2	7	12	5	10	3	8
Alternate ccw	1	8	6	3	11	10	4	5	9	12	2	7
Alternate cw	1	6	8	11	3	4	10	9	5	2	12	7

Table 1: Blue denotes consonant pole, yellow denotes dissonant pole.

In another mapping approach (Fig 3), values were taken from top to bottom, alternating from left to right. In this mapping values between 1 and 12 were individually assigned to their respective dissonant/consonant values, which then were used to specify a tone to be played.

The octave in which to play a tone was dependent on the data category. For the case of time spent at constant velocity the octave was from A3 to A4. For the case of time spent at average direction the octave was A4 to A5.

4.6. Tempo and Key

After the Flow Graph specifies an octave and tone to play, it then specifies a time at which to play. This has been mapped proportionally to a body's velocity. The faster a body is moving, the faster the tempo and hence shorter the note duration.

The transposition to an appropriate key (music scale) was important for this design since the key determines the root note and consequently the consonant and dissonant poles. The background music of a supplementary video to the interactive exhibition was in the key of A (220 Hz). The sound library (as discussed in 4.1) was transposed to the key of A, such that tone 1 became A and tone 7 became D. The dissonance and consonance could be heard from the "baseline" of the background music.

4.7. Gallery Installation

Two webcams were placed on the ceiling in order to reduce occlusion of marked bodies and increase visible floor area. Markers were attached to the body of a wheelchair and scooter. The markers used were 2D graphics printed on white paper, each unique in pattern but identical in scale (this was to ensure identical scaling in game). The marker tracking application returned x, y, z coordinates for both translation and rotation, which were written to an XML file as streaming data.

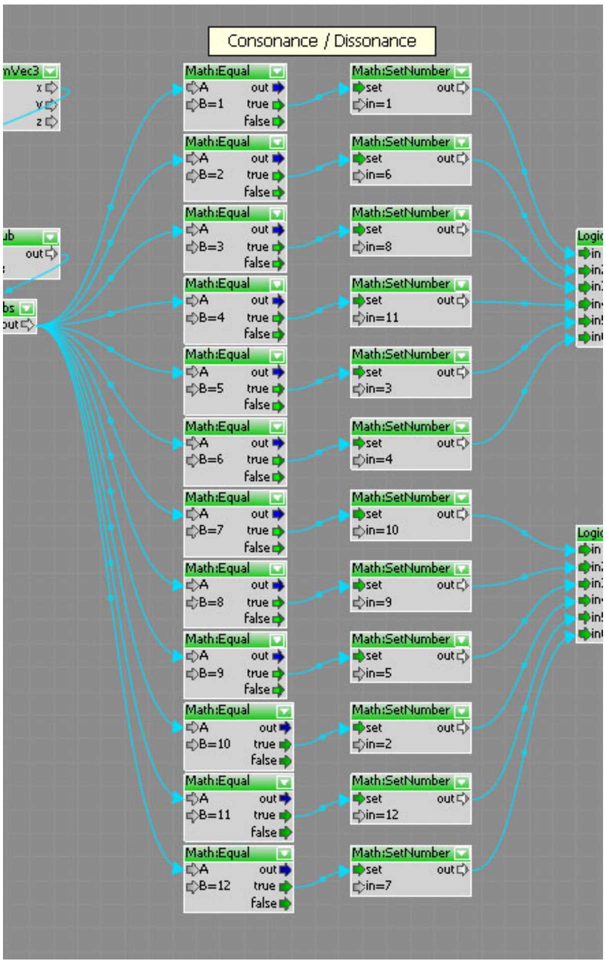

Figure 3: Flow Graph: nominal linking of alternating clockwise mapping.

As the marked bodies were moved by visitors, the XML file would be updated accordingly, which when harvested by the Flow Graph in Crysis Wars Sandbox 2 would similarly move a representation of the body on screen.

This action caused the visualization and sonification to change in real-time. The gallery space became a sonic environment controlled by the visitor and their interactivity with the bodies. (Fig. 4)

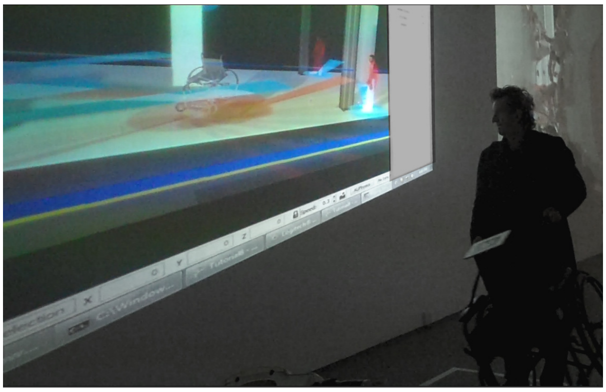

Figure 4: Wheelchair object articulated on screen.

5. DISCUSSION

The implementation of a computer game engine through the use of its editor will be discussed here with a focus on the opportunities and limitations it affords. The design case and interactive exhibition allowed testing and evaluation. The discussion covers areas of flexibility and accessibility, synchronization of visualisation and sonification, sound synthesis and music composition, distribution, recording and retrieving data (xml), sonification of external data sets, sonification of multiple data sets simultaneously and maintaining data integrity

The Flow Graph within Crysis Sandbox 2 made the computer game editor more accessible than other existing of programming environments designed for sound production and sonification, since it did not require any code knowledge or code keywords. As a tool for sonification, Crysis Sandbox 2 was suited to designers, since the Flow Graph encouraged a design-orientated approach. The limitation of this feature is that some of the Flow Graphs created for simple tasks, such as storing variables, are more complex than they need to be and place unnecessary processing load on a computer. The same module written in code would be more efficient. Crysis Sandbox 2 allows custom extension by C++ and LUA scripting. The flexibility of the parametric mappings gives a designer the opportunity to interact with the sonification. In this design case the DisCo mappings were tested by changing parameters within the Flow Graph. This interaction could theoretically be extended to end users/analysts who could change parameters in real-time as they listen to a sonification of streaming data.

The synchronization of visualization and sonification was made possible by the implementation of a computer game editor. The Flow Graph was able to articulate changes in tone and octave whilst articulating changes in color, scale, direction, translation of 3D objects. The design case demonstrated the opportunity to interact with synchronized 3D visualization and sonification in real-time.

The advantage of game engines to sonify streaming data in real-time come at the cost of synthesizing sinusoids into sounds. When using a game engine editor for sonification, the base generation of sound is not performed in real-time. Instead sounds are pre-processed into a sample library which is used to compose samples together in real-time depending on the variables of a data stream. This inability to synthesis signals of sinusoids into complex sound is a limitation. However, there is an opportunity for real-time composition of sound samples.

Distribution of sonification mappings and their results have limitations and opportunities. The output format of a sonification result from a gaming engine can be recorded as video and/or audio which is distributable but in a sense “frozen.” The ability of Walker’s sonification sandbox to output MIDI offers advantages that the gaming engine method does not, because MIDI can be played on various platforms and programs. For the distribution of mappings, the ability to export and import Flow Graphs makes sharing parametric algorithms or components of them possible. For the distribution of whole levels, the folder structure of a custom game mod allows itself to be shared. These last two forms of distributions are effective methods since they will maintain the interactivity of the sonification design for another user.

The XML file format used with Crysis Wars Sandbox 2 is similar to the CSV file format used with Walker’s sonification sandbox [5]. Rather than import a data set, the implementation of a game engine editor allows periodic harvesting of a data file which enables real-time interaction with streaming data. The use of XML file format also allows new data to be recorded, archived and retrieved. For example extrapolated data that is calculated by a game engine from multiple files can be recorded to a new file. This form of data recording can be seen as creating internal data sets and is similar to LeGroux’s VR-RoBoser [10].

In addition to handling internal or self-created data, the possibility of sonifying external data sets is demonstrated in the design case within this paper where the positional data captured from camera and marker tracking is written in real-time to XML format, then translated and mapped in the computer game editor.

The ability to read multiple XML files and combine data from several sources is another opportunity that game engines offer. This multiple file harvesting was utilized in this design case where two cameras were used to increase the available tracking area. An XML was created and populated for each camera and the Flow Graph was used to harvest data from each XML. Extending from this ability to read multiple data sets simultaneously, there is also potential to overlay data from recordings of different events, including prerecorded video.

The advantage of harvesting data from an external source ensures that raw data was not altered but only translated and represented in game. The mapping of data in the design case outlined in this paper maintains relational data integrity but compromises numerical data integrity. Imposing a structure of music through employing dissonant and consonant mappings enables an ability to represent similar yet different data relationships simultaneously within the dimension of pitch. In this case two different objects moving at different velocities could sound consonant if they had remained at their respective velocities for a similar duration of time.

The granularity of a musical structure distorts the accuracy of absolute data values and therefore would not suit applications where the values themselves are important. For applications where relations between data are of interest, then this method of mapping is useful. The loss of resolution is compensated by an increase in expression of data relationships. Rather than conveying data on a high resolution linear spectrum, this design case conveyed data on a granular spectrum that expressed interrelationships.

6. CONCLUSION

It has been demonstrated that a computer game engine can be implemented as a sonification tool through the use of its editor/sandbox. This has been shown to offer some opportunities over existing tools, but not without some limitations.

Opportunities include the ability to record, archive and retrieve data in XML file format, with the potential to retrieve data from multiple data sets simultaneously; distribution while maintaining interactivity through sharing Flow Graphs or custom game mod folders; and flexibility and accessibility through a graphical programming interface that does not require code knowledge.

This paper focused on the manipulation of pitch and its mappings, and to a lesser extent demonstrated an ability to alter tempo and tone duration. It did not however modify volume/amplitude, and the more inherently potent spatial dynamics of which a computer game is capable. The limitations of implementing a computer game engine include the inability to synthesis sinusoids into sounds. However it is likely that game engine developers will offer such capabilities as processing technology advances.

If game engines continue to offer capabilities such as graphic programming, real-time calculations of incoming data, distributable mods and graphs, ability to retrieve data from multiple sources, then they will continue to be useful for sonification. There is an opportunity to implement game engines for sonification purposes in areas/applications which already use game engines as a tool for visualization. These include educational games, exercise and health games, medical and healthcare training/simulation, defense training, emergency and evacuation management, town planning and architecture representation.

If sonification, as identified by Kramer et al., has similarities to the developments in visualization, and if sonification is an additional dimension to vision, through which to convey information - then we should look at not only tools designed for sonification and sound synthesis that have an ability to produce graphics, but equally tools designed for advanced real time 3d visualization that have the ability to produce sound. This will enable movement towards suitable tools and toolkits that are accessible to designers seeking to experiment on the partnership of visual and aural representation.

7. FUTURE WORK

The direction of this research is towards a scalable design, in which the movement of bodies may increase to the movement of groups of bodies. Currently this work invites active interaction due to its transparency as a reactive art exhibition, however it is envisaged that an investigation into a passive interaction system would invite different movement and thus a different data set.

An aspect of the gaming engine that this research did not exploit was the ability to convey sound spatially. Whether this eventuates as stereo or multi-point surround, it will add to the dimensionality of its potential as a sonification system.

This paper has isolated and explored the tonal mappings of pitch, in the granular form of dissonance and consonance. Further development will see a focus directed towards implementing a computer game editor to manipulate other variables of sound and music such as tempo, volume, attack and timbres.

8. ACKNOWLEDGMENT

The research is jointly funded by the Australian Research Council (ARC LP 0991589), the University of New South Wales (UNSW), and the Emergency Information Coordination Unit (EICU).

The author wishes to thank Richard Goodwin and Russell Lowe, whose exhibition in Beijing included the contributions of the sonification design outlined in this paper.

9. REFERENCES

[1] G. Kramer, et al., "Sonification Report: Status of the Field and Research Agenda," 1997. http://www.icad.org/websiteV2.0/References/nsf.html [accessed 28/2/2011]

[2] C. M. Wilson and S. K. Lodha, "Listen: A Data Sonification Toolkit," *The 3rd International Conference on Auditory Display*, Palo Alto, USA, 1996.

[3] S. K. Lodha, et al., "MUSE: A Musical Data Sonification Toolkit," *The 4th International Conference on Auditory Display*, Palo Alto, USA, 1997.

[4] A. J. Joseph and S. K. Lodha, "MUSART - Musical Audio Transfer Function Real-Time Toolkit," *The 8th International Conference on Auditory Display*, Kyoto, Japan, 2002.

[5] B. N. Walker and J. T. Cothran, "Sonification Sandbox: A Graphical Toolkit for Auditory Graphs," *The 9th International Conference on Auditory Display*, Boston, USA, 2003.

[6] J. Guscott, "Colourful Music: The Interactive Soundscape of 'de Blob,'" *Game Connect: Asia Pacific*, Melbourne, Australia, 2009.

[7] A. Hunt and T. Hermann, "The Importance of Interaction in Sonification," *The 10th International Conference on Auditory Display*, Sydney, Australia, 2004.

[8] A. Hunt, et al., "Interacting with Sonification Systems: Closing the Loop," in *Proceedings of Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, 2004, pp. 879-884.

[9] M. Grimshaw, "Sound and Immersion in the First-Person Shooter," *International Conference on Computer Games (CGames)*, La Rochelle, France, 2007.

[10] S. LeGroux, et al., "VR-RoBoser: Real-Time Adaptive Sonification of Virtual Environments Based on Avatar Behavior," in *Proceedings of Conference on New Interfaces for Musical Expression*, New York, USA, 2007, pp. 371-374.

[11] H. Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, Longmans, Green, 1912. Original publication *Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik, 1863.*

[12] J. Cage, *Indeterminacy: New Aspect of Form in Instrumental and Electronic Music. Ninety Stories by John Cage, with Music.* John Cage, reading; David Tudor, music, 1959.

[13] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition*, Pendragon Press, 2001 (original 1955).

[14] J. L. Alty, "Can We Use Music in Computer-Human Communication?," in *People and Computers X: Proceedings of HCI '95*, M. A. R. Kirby, A. J. Dix and J. E. Finlay, Eds. Cambridge, Cambridge University Press, 1995, pp. 409–423.

[15] P. Vickers and J. L. Alty, "Musical Program Auralisation: Empirical Studies," *The 6th International Conference on Auditory Display*, Atlanta, USA, 2000, pp. 157-166.

[16] D. Worrall, "Parameter Mapping Sonic Articulation and the Perceiving Body," *The 16th International Conference on Auditory Display*, Washington D.C., USA, 2010.

[17] J. Kursell, "Hermann von Helmholtz and Carl Stumpf on Consonance and Dissonance," *Berichte zur Wissenschaftsgeschichte*, vol. 31, no. 2, pp. 130-143, 2008.

[18] C. Stumpf, Konsonanz und Dissonanz, Berichte zur Akustik und Musikwissenschaft 1, 1898.

[19] R. Plomp and W. J. M. Levelt, "Tonal Consonance and Critical Bandwidth," *Journal of the Acoustical Society of America*, vol. 38, no. 4, 1965, pp. 548-560.

[20] S. Hinton, "The Emancipation of Dissonance: Schoenberg's Two Practices of Composition," *Music and Letters*, vol. 91, no. 4, 2010, pp. 568-579. Hinton cites *Harmonielehre* 1911, translated to English by Roy Carter as *Theory of Harmony*, 1922.

[21] C. R. Jensen, "A Theoretical Work of Late Seventeenth-Century Muscovy: Nikolai Diletskii's "Grammatika" and the Earliest Circle of Fifths," *Journal of the American Musicological Society*, vol. 45, no. 2, 1992, pp. 305-331.

[22] "Sandbox 2 Manual: Flowgraph Editor," in *CryEngine 2 Documentation*, http://doc.crymod.com/SandboxManual [accessed 28/2/2011]

[23] S. A. Malinowski, *Harmonic Coloring: A Method for Indicating Pitch Class*, http://www.musanim.com/mam/circle.html [accessed 28/2/2011]

[24] S. S. Stevens, "On the Theory of Scales of Measurement," *Science*, vol. 103, no. 2684, 1946, pp. 677-680.

[25] J. Zhang, "A representational analysis of relational information displays," *Int. J. Hum.-Comput. Stud.*, vol. 45, no. 1, 1996, pp. 59-74.